The Relational Model 3: Design

- Decomposition
  - Lossless Join
  - Projecting FDs
  - Minimal Basis for FDs
- Decomposition into BCNF
- 3NF

Readings: Sections 3.2.8, 3.3—3.5 of Textbook

BCNF Revisited

- When designing a relational database schema, BCNF is the most desirable form for relations.
  - A relation is in BCNF if for any nontrivial FD $X \rightarrow Y$ in R, X is a superkey of R.
  - The stepts to check the goodness of a relation schema:

    FDs ... Superkeys ... BCNF

Excercise

Consider the Class relation about the timetable for a university and FDs:

Class(cno, title, prog, progtype, progleader, day, time, room, classtype, capacity)

cno → title, prog
prog → progtype, progleader
day, time, room → cno, classtype
room → classtype, capacity

- cno, title: number. and title of a course;
- prog, progtype, proglader: title, type (e.g. postgraduate), and leader of a program;
- day, time, room: scheduled time and venue for a classes;
- classtype: class type (lecture, tute, lab);
- capacity: the capacity of a room.

What are the superkeys of Class? Is Class in BCNF?

Decomposition

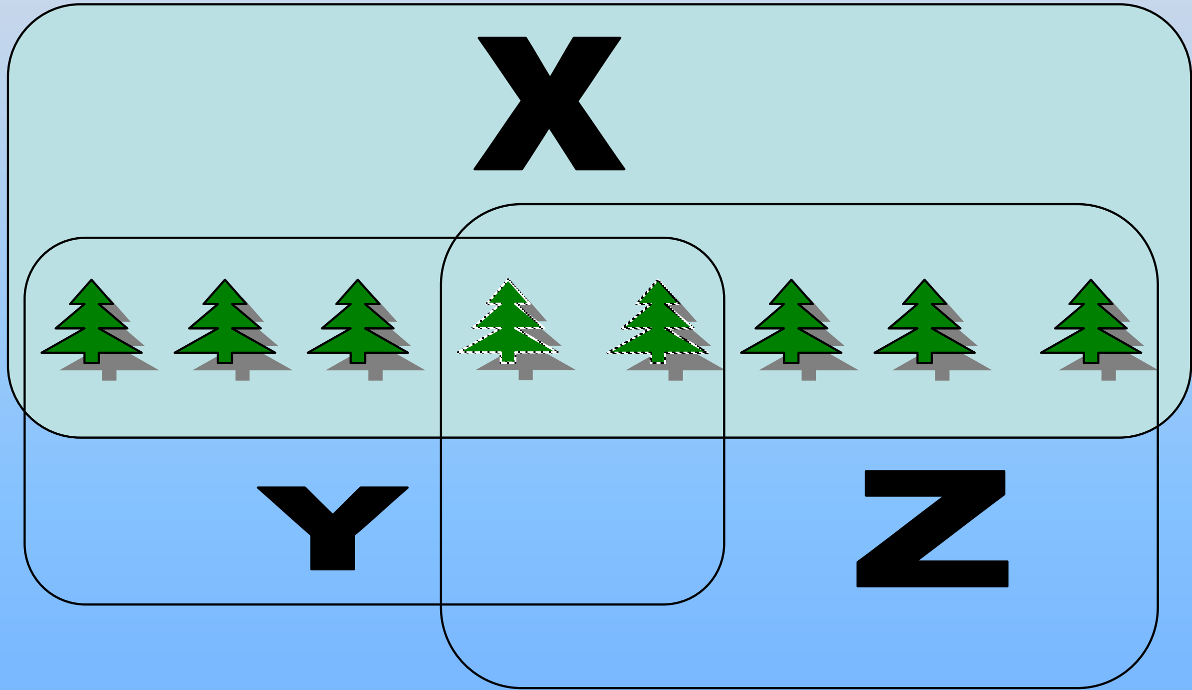
- A relation not in BCNF needs to be decomposed into relations in BCNF.
- Some necessary concepts for correct decomposition.
  - Decomposition properties:
    - Lossless join
    - Dependency preservation
  - Minimal basis for FDs

Decomposition

- A relation schema $R(X)$ is decomposed into two relations $R_1(Y)$ and $R_2(Z)$, where X, Y, Z are sets of attributes and $X = Y \cup Z$.
  - Each attribute in X must be included at least once in either X or Y.
  - Some attributes in X may be included in both Y and Z.
  - Tuples of instance for R are projected onto attributes in Y and Z respectively.

union

Decomposition ...



$X = Y \cup Z$

Decomposition: Example

Class

cno	title	day	time	room	type
isys1057	Database Concepts	Wed	1030	14.04.27	lect
isys1057	Database Concepts	Thur	1330	14.04.27	lect
isys1055	Database Concepts	Wed	1730	12.05.02	lect
isys1057	Database Concepts	Wed	1130	14.10.30	tute
isys1057	Database Concepts	Wed	1330	14.09.23	tute
acct1009	Another Course	Wed	1130	14.04.27	lect
acct1009	Another Course	Thur	1330	07.02.23	lect
acct1009	Another Course	Thur	1430	07.02.23	tute

Tuples projected to {cno, title}.

Tuples projected to {cno, day, time, room, type}.

ClassInfo

cno	title
isys1057	Database Concepts
isys1055	Database Concepts
acct1009	Another Course

CourseClass

cno	day	time	room	type
isys1057	Wed	1030	14.04.27	lect
isys1057	Thur	1330	14.04.27	lect
isys1055	Wed	1730	12.05.02	lect
isys1057	Wed	1130	14.10.30	tute
isys1057	Wed	1330	14.09.23	tute
acct1009	Wed	1130	14.04.27	lect
acct1009	Thur	1330	07.02.23	lect
acct1009	Thur	1430	07.02.23	tute

Decomposition – Lossless Join

- Relation R is decomposed into R_1 and R_2. Tuples in an instance for R can be reconstructed from rejoining the projected tuples for R_1 and R_2 on common attributes. The decomposition is a lossless-join decomposition, or simply lossless.
  - R_1 and R_2 must at least have common attributes.
  - For any instance of R, joining tuples in R_1 and R_2 on common attributes produces rightful tuples for R.
  - For any instance of R, joining tuples in R_1 and R_2 on common attributes does not produce spurious tuples not in R.

Lossless/Lossy Join: Example

Class(cno, title, day, time, room, type)

cno → title

day, time, room → cno, type

room → type

Common attribute cno

- The following decomposition is lossless.

  CourseInfo(cno, title)

  CourseClass(cno, day, time, room, type)

- The following decomposition is lossy.

  Class1(cno, title, type)

  Class2(day, time, room, type)

Common attribute Type

Lossless Join: Example ...

cno	title
isys1057	Database Concepts
isys1055	Database Concepts
acct1009	Another Course

cno	day	time	room	type
isys1057	Wed	1030	14.04.27	lect
isys1057	Thur	1330	14.04.27	lect
isys1055	Wed	1730	12.05.02	lect
isys1057	Wed	1130	14.10.30	tute
isys1057	Wed	1330	14.09.23	tute
acct1009	Wed	1130	14.04.27	lect
acct1009	Thur	1330	07.02.23	lect
acct1009	Thur	1430	07.02.23	tute

Join on common attribute Course

cno	title	day	time	room	type
isys1057	Database Concepts	Wed	1030	14.04.27	lect
isys1057	Database Concepts	Thur	1330	14.04.27	lect
isys1057	Database Concepts	Wed	1130	14.10.30	tute
isys1057	Database Concepts	Wed	1330	14.09.23	tute
isys1055	Database Concepts	Wed	1730	12.05.02	lect
acct1009	Another Course	Wed	1130	14.04.27	lect
acct1009	Another Course	Thur	1330	07.02.23	lect
acct1009	Another Course	Thur	1430	07.02.23	tute

Lossy Join: Example

cno	title	type
isys1057	Database Concepts	lect
isys1055	Database Concepts	lect
isys1057	Database Concepts	tute
acct1009	Another Course	lect
acct1009	Another Course	tute

day	time	room	type
Wed	1030	14.04.27	lect
Thur	1330	14.04.27	lect
Wed	1730	12.05.02	lect
Wed	1130	14.10.30	tute
Wed	1330	14.09.23	tute
Wed	1130	14.04.27	lect
Thur	1330	07.02.23	lect
Thur	1430	07.02.23	tute

Join on common attribute Type

cno	title	day	time	room	type
isys1057	Database Concepts	Wed	1030	14.04.27	lect
isys1057	Database Concepts	Thur	1330	14.04.27	lect
isys1057	Database Concepts	Wed	1730	12.05.02	lect
isys1057	Database Concepts	Wed	1130	14.04.27	lect
isys1057	Database Concepts	Thur	1330	07.02.23	lect
isys1055	Database Concepts	Wed	1030	14.04.27	lect
isys1055	Database Concepts	Thur	1330	14.04.27	lect
isys1055	Database Concepts	Wed	1730	12.05.02	lect
isys1055	Database Concepts	Wed	1130	14.04.27	lect
isys1055	Database Concepts	Thur	1330	07.02.23	lect
isys1057	Database Concepts	Wed	1130	14.10.30	tute
isys1057	Database Concepts	Wed	1330	14.09.23	tute
isys1057	Database Concepts	Thur	1430	07.02.23	tute
acct1009	Another Course	Wed	1030	14.04.27	lect
acct1009	Another Course	Thur	1330	14.04.27	lect
acct1009	Another Course	Wed	1730	12.05.02	lect
acct1009	Another Course	Wed	1130	14.04.27	lect
acct1009	Another Course	Thur	1330	07.02.23	lect
acct1009	Another Course	Wed	1130	14.10.30	tute
acct1009	Another Course	Wed	1330	14.09.23	tute
acct1009	Another Course	Thur	1430	07.02.23	tute

isys1057 should have two lectures but now has 5 lectures!!!

Spurious tuples !!!

Lossless Join Decomposition Process

- Decomposed relations must at least have common attributes --- not enough!
  - The common attributes must be a key in at least one relation after decomposition.
- Examples:
  - Lossless Join -- What are the keys for the following relations?

    CourseInfo(cno, title)

    CourseClass(cno, day, time, room, type)

  -- Lossy decomposition -- What are the keys for the following relations?

    Class1(cno, title, type)

    Class2(day, time, room, type)
- The Chase test for lossless Join --- see Sections 3.4.2 and 3.4.3 of Textbook for details

Decomposition – Projecting FDs

- When a relation R is decomposed into R_1 and R_2, FDs in R are projected onto R_1 and R_2.
- Issues to consider:
  - How to find the FDs on R_1 and R_2?
  - The FDs in R_1 and R_2 may have redundancy. How to construct a minimal set of these FDs?
  - Can all FDs in R hold on R_1 and R_2?

Projecting FDs

- In principle, a set of FDs *S* projected onto a set of attributes R_1 should include FDs that
  - Are inferred from *S*;
  - Involve only attributes in R_1.
- Example: Given a set of FDs on student information:

  cno → title, prog

  prog → progtype, progleader

  day, time, room → cno, classtype

  room → classtype, capacity

  staff → classtype

  - Find redundancy in the above set of FDs.
  - What are the FDs projected onto {cno, progleader}?

  cno → progleader! Why?

Projecting FDs

Input: A relation R, a set of FDs S holding in R, and R_1 – a subset of R.

Output: the set of FDs that holds in R_1.

1. Let T be the set of FDs on R_1. Initialise T to empty set.
2. For each attribute X in R_1, compute X^+ w.r.t. S. Add to T all non-trivial FDs $X \rightarrow A$ such that A is in X^+ and R_1.
3. Now T contains all FDs that holds on R_1 but may contain redundancy -- find a minimal basis (explained later).

Projecting FDs: Example

Consider the following FDs and a subset attributes

R_1={cno, title, progtype}:

cno → title, prog

prog → progtype, progleader

day, time, room → cno, classtype

room → classtype, capacity

What are the (minimal) FDs projected on R_1?

- {cno}+={title, prog, progtype, progleader}. Non-trivial FDs Derived that are like "cno → *" and projected onto R_1:

  cno→ title; cno→ progtype
- {title}+={title}. Non-trivial FDs "title → *" onto R_1: Nil.
- {progtype}+? Non-trivial FDs "progtype→ *" projected onto R_1?

The resulting FDs projected onto R_1:

cno→ title

cno→ progtype

Minimal Basis for FDs

- Redundant FDs: FDs that can be inferred from other FDs.
- Redundant attributes in FDs: some attributes on the left hand side of FDs can be removed.
- A minimal basis for FDs is equivalent to the FDs but without redundancy:
  - Right sides are single attributes.
  - No FD can be removed.
  - No attribute can be removed from left hand side.

Minimal Basis for FDs: Example

Continuing with slide #14, redundant FDs:

- course→ progtype
- course→ progleader

The minimal basis for FDs on {course, title, prog, progtype, progleader)

- course → title
- course→ prog
- prog→ progtype
- prog→ progleader

Finding a Minimal Basis for FDs

1. Replace an FD with multiple attributes on the right with FDs with single attribute on the right.
2. Remove an FD if it can inferred from other FDs.
3. Remove an attribute from the left side of an FD if the FD after such removal is equivalent to the original.
4. Repeat steps 2 and 3 until no FD or attribute can be removed.

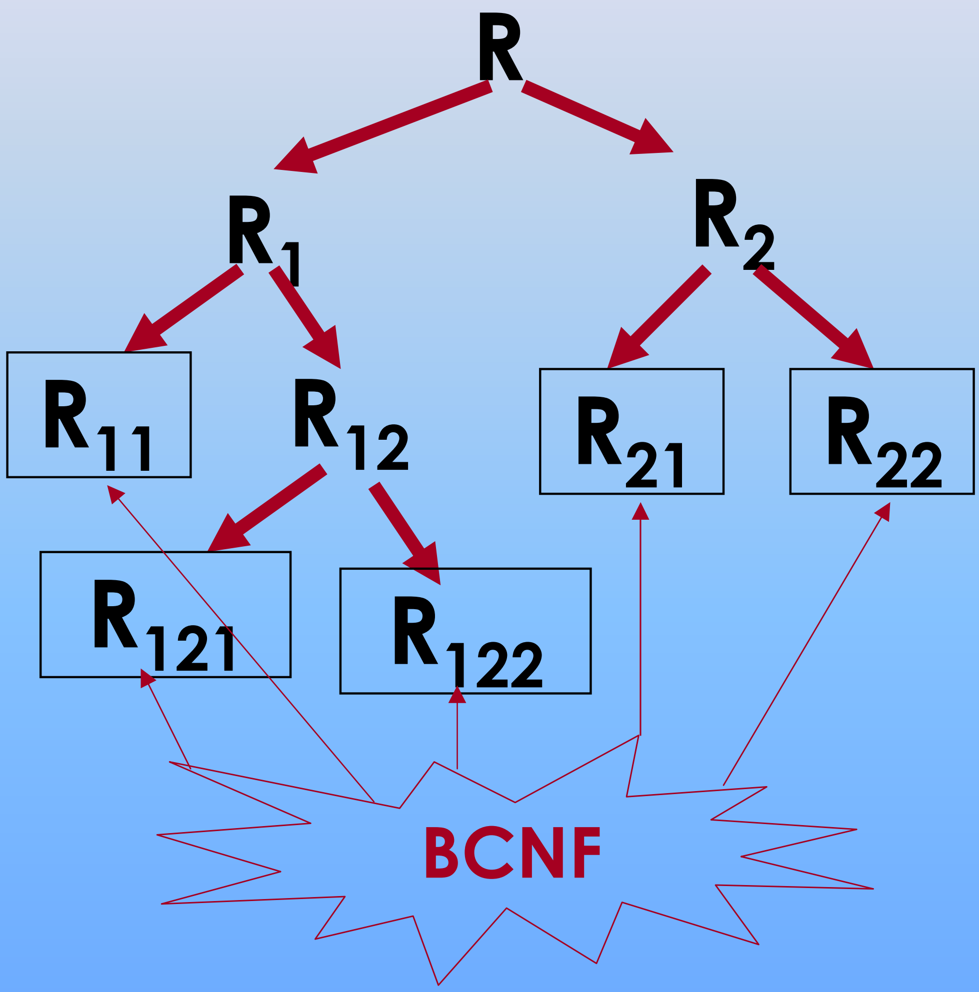
Decomposition into BCNF

Input: a relation with FD's but not in BCNF.

Output: a set of relations in BCNF.

1. Check given FD's for a BCNF violation $X \rightarrow Y$.
2. Decompose the relation using $X \rightarrow Y$ (see the next slide) into R_1 and R_2.
3. Repeat Steps 1 and 2 on R_1 and R_2 respectively until there are not any BCNF violation FDs in any relations.

Decomposition into BCNF ...



R is decomposed into 5 relations and they are all in BCNF.

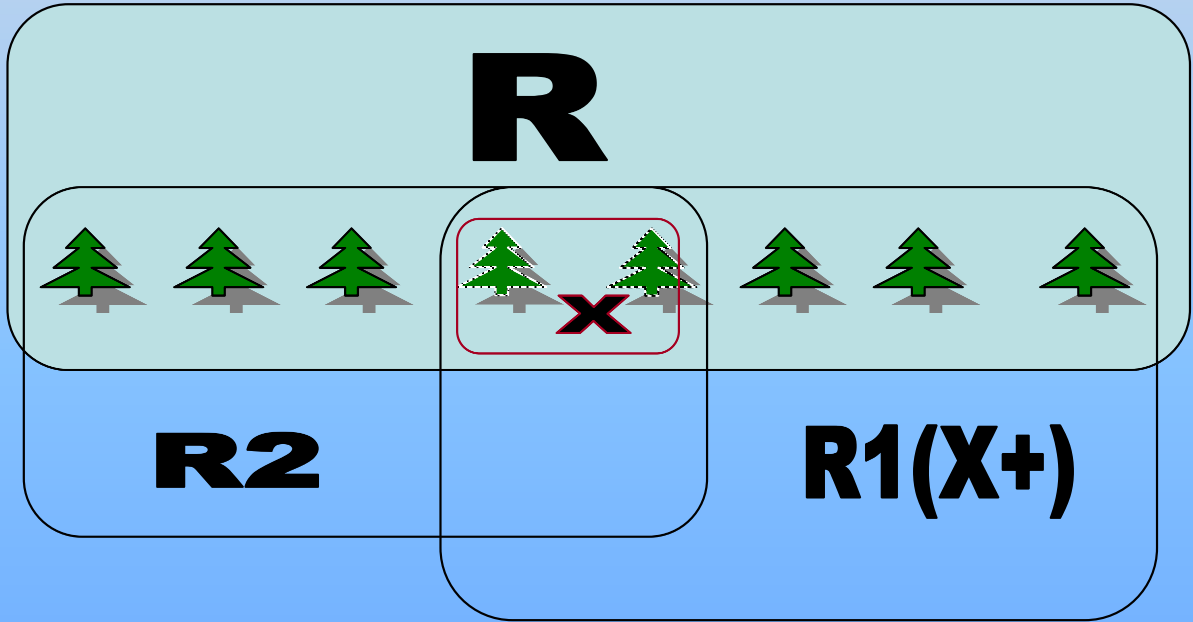
Decompose R Using $X \rightarrow Y$

- Replace R by relations with schemas:

  $R_1 = X^+$.

  $R_2 = R - (X^+ - X)$.

  Note that X^+-X are the new attributes dependent on X.
- Project given FD's F onto the two new relations.

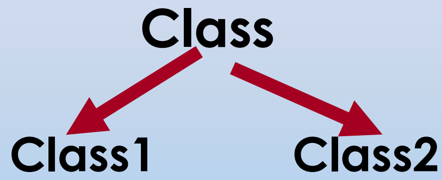


X is the set of common attributes between R_1 and R_2, and X is the key for R_1.

Example: the Class Relation

Class(cno, title, prog, progtype, progleader, day, time, room, classtype, capacity)

cno → title, prog
prog → progtype, progleader
day, time, room → cno, classtype
room → classtype, capacity



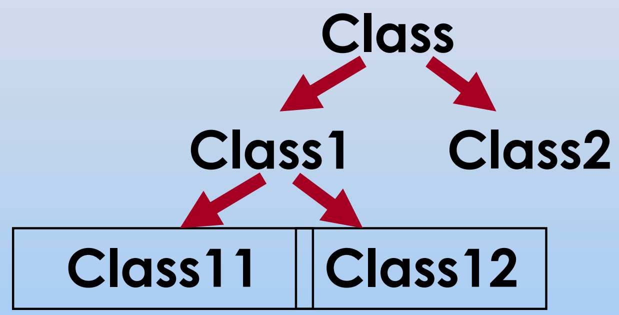
The key of Class is {day, time, room}. Class is not in BCNF. BCNF violation FDs:

cno → title, prog
prog → progtype, progleader
room → classtype, capacity

Decompose Class using cno → title, prog

1. Closure of the left hand side:
   $\{cno\}^+$ = {cno, title, prog, progtype, progleader}.
2. Decompose Class into:
   Class1(cno, title, prog, progtype, progleader)
   Class2(cno, day, time, room, classtype, capacity)

Example ...



Project FDs onto Class1 (a subset of attributes):

Class1(cno, title, prog, progtype, progleader)

- cno → title
- cno → prog
- prog→ progtype
- prog → progleader

{cno} is a (super)key for Class1 but {prog} is not. Class1 is not in BCNF. BCNF violations:

prog → progtype

prog → progleader

Decomposition of Class1 using prog→ progtype:

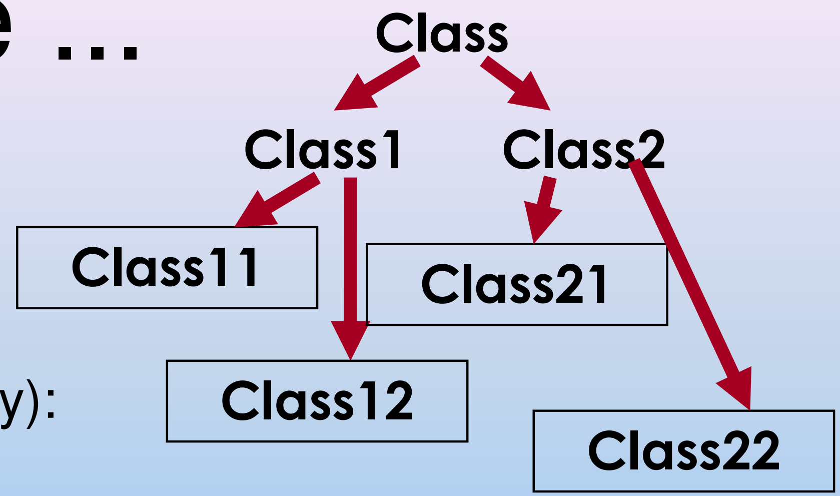
1. Compute $\{prog\}^+$ = {prog, progtype, progleader}.
2. Decomposition:

   Class11(prog, progtype, progleader)

   Class12(cno, title, prog)

The keys are underlined. Both Class11 and Class22 are in BCNF.

Example ...



- Class2(cno, day, time, room, classtype, capacity):
  - day, time, room → cno, classtype
  - room → classtype,capacity
- The key for Class2 is {day, time, room}. Class2 is not in BCNF. BCNF violations:
  - Room → classtype
  - Room → capacity

  Decompose of Class2 using room → classtype

  Compute $\{room\}^{+}$={room, classtype, capacity}

  Decomposition:
  - Class21(room, classtype, capacity}
  - Class22(room, day, time, cno}

  The keys are underlined. Class21 and Class22 are both in BCNF.

Example ... Summary

To recap, the Class relation with a schema as follows and not in BCNF:

Class(cno, title, prog, progtype, progleader, day, time, room, classtype, capacity)

is decomposed into 4 relations in BCNF:

- Class11(prog, progtype, progleader)
- Class12(cno, title, prog)
- Class21(room, classtype, capacity}
- Class22(room, day, time, cno}

Each of the 4 relations is talking about one thing:

- Class11: program info.
- Class12: course info and course-program relationship.
- Class21: room info.
- Class22: classes for courses.

Unenforceable FDs

- Ideally decomposition into BCNF should satisfy:
  - Lossless join: decomposed relations should be joined to produce the original relation.
  - FD preservation: all FDs hold in the original relation should also be enforced in the decomposed relations.
    - But this sometimes can not be achieved – some FDs are unenforceable in decomposition.

Unenforceable FD …

Given Lecture(room, course, staff) and FDs:

room, course → staff

staff → course

Lecture has two keys:

{room, course}

{room, staff}

Lecture is not in BCNF. staff → course is a BCNF violation. Following the decomposition into BCNF process, we decompose Lecture using staff → course:

Teach(course, staff): staff → course

Lect2(room, staff): Nil FDs.

It is impossible to enforce FD room, course → staff in the decomposed relations!

Unenforceable FDs …

- The FD room, course → staff is not enforceable and it is also problematic. Data causing the FD violation can be added into the decomposed relations.
- So, the relation Leture(room, course, staff) should not be decomposed.

course	staff
KDW	JZ
KDW	IB

Staff → course

room	staff
14.10.30	JZ
14.09.14	JZ
14.09.14	IB

Join on common attribute Staff

room	course	staff
14.10.30	KDW	JZ
14.09.14	KDW	JZ
14.09.14	KDW	IB

FD
room, course → staff
is violated.

Solution — 3NF

- Third Normal Form (3NF) loosens the BCNF condition so we do not have to decompose in this problem situation.
- A relation R is in 3NF if for any FD *X ->A on R, X is a set of attributes and A is an attribute, either X is a superkey or A is part of a key (maybe a different key).*
  - When applying the definition on minimal FDs, it is that either X is a key or A is part of a key.

3NF: Example

- Lecture(room, course, staff)
  - FD1: Room, course → staff
  - FD2: Staff → course
- There are two keys

  {room, course} and {staff, room}.

  FD1: the left is a key.

  FD2: the left is not a key, but Course on the right is part of the key {room, course}.

Lecture is not in BCNF, but in 3NF.

Decomposition into 3NF

Decomposition into 3NF relations with a lossless join and dependency preservation.

1. Construct a minimal basis for given FDs.
2. One relation for each FD in the minimal basis.
   – Schema is the union of the left and right sides.
3. If no key of the original relation is contained in an FD, then add one relation whose schema is some key.

Proof that the process is lossless join and dependency preserving and produces 3NF relations is in Section 3.5.3 of Textbook.

Example

Let us revisit decomposition of the Class relation:

Class(cno, title, prog, progtype, progleader, day, time, room, classtype, capacity)

cno → title, prog
prog → progtype, progleader
day, time, room → cno, classtype
room → classtype, capacity

1. The minimal basis for FDs:

cno → title
cno → prog
prog → progtype
prog → progleader
day, time, room → cno
room → classtype
room → capacity

The key for Class is {day, time, room}.

Example …

2. Constructing relations:
   - R1(cno, title)
   - R2(cno, prog)
   - R3(prog, progtype)
   - R4(prog, progleader)
   - R5(day, time, room, cno)
   - R6(room, classtype)
   - R7(room, capacity)

Combining relations:
   - Class1(cno, title, prog*)
   - Class2(prog, progtype, progleader)
   - Class3(day, time, room*, cno*, classtype*)
   - Class4(room, classtype, capacity)

3. The Key of the original Class relation is in relation Class3 and no additional relation is needed.

The resulting relations are the same as those from Decomposition into BCNF. The relations are in 3NF, and indeed also in BCNF.

Decomposition into BCNF/3NF

- Lossless Join (LJ) and Dependency Preservation (DP) are desirable properties of decomposition into BCNF/3NF.
- The BCNF decomposition (Slide #18) ensures LJ.
- The 3NF decomposition (Slide #30) ensures LJ and DP.
- The two decompositions often result in the same relations.
  - The BCNF decomposition often keeps all FDs.
  - The 3NF decomposition often results in BCNF relations.
- Use 3NF decomposition is preferred.

Exercise

Given relation $R(A, B, C, D)$ with FDs

$A \rightarrow B$, $A \rightarrow D$

What normal form is R in? If not in BCNF/3NF, decompose R into BCNF/3NF relations.